HHS Public Access
Author manuscript
Am Stat. Author manuscript; available in PMC 2019 November 14.

Published in final edited form as:
Am Stat. 2018 ; 72(4): 382–391. doi:10.1080/00031305.2017.1356747.

A Guide to Teaching Data Science

Stephanie C. Hicks[1,2] and **Rafael A. Irizarry**[1,2]

[1]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA

[2]Department of Biostatistics, Harvard School of Public Health, Boston, MA

Abstract

Demand for data science education is surging and traditional courses offered by statistics departments are not meeting the needs of those seeking training. This has led to a number of opinion pieces advocating for an update to the Statistics curriculum. The unifying recommendation is that computing should play a more prominent role. We strongly agree with this recommendation, but advocate the main priority is to bring applications to the forefront as proposed by Nolan and Speed (1999). We also argue that the individuals tasked with developing data science courses should not only have statistical training, but also have experience analyzing data with the main objective of solving real-world problems. Here, we share a set of general principles and offer a detailed guide derived from our successful experience developing and teaching a graduate-level, introductory data science course centered entirely on case studies. We argue for the importance of *statistical thinking*, as defined by Wild and Pfannkuch (1999) and describe how our approach teaches students three key skills needed to succeed in data science, which we refer to as *creating*, *connecting*, and *computing*. This guide can also be used for statisticians wanting to gain more practical knowledge about data science before embarking on teaching an introductory course.

Keywords

data science; applied statistics; teaching principles; active learning; reproducibility; computing

1. INTRODUCTION

1.1 What do we mean by *Data Science*?

The term *data science* is used differently in different contexts since the needs of data-driven enterprises are varied and include acquisition, management, processing, and interpretation of data as well as communication of insights and development of software to perform these tasks. Michael Hochster[1] defined two broad categorizations of data scientists: "*Type A [for 'Analysis'] Data Scientist*" and "*Type B [for 'Building'] Data Scientist*", where Type A is "*similar to a statistician… but knows all the practical details of working with data that aren't taught in the statistics curriculum*" and Type B are "*very strong coders … may be trained as*

[1] https://www.quora.com/What-is-data-science

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

software engineers and mainly interested in using data 'in production". Here we focus on the term *data science* as it refers generally to Type A data scientists who process and interpret data as it pertains to answering real-world questions. We do not make any recommendations as it pertains to training Type B data scientist as we view this as a task better suited for engineering or computer science departments.

1.2 Why are statistics[2] departments a natural home for Data Science in Academia?

Current successful Data Science education initiatives in academia have resulted from combined efforts from different departments. Here we argue that statistics departments should be part of these collaborations. The statistics discipline was born directly from the endeavour most commonly associated with data science: data processing and interpretation as it pertains to answering real world questions. Most of the principles, frameworks and methodologies that encompass this discipline were originally developed as solutions to practical problems. Furthermore, one would be hard pressed to find a successful data analysis by a modern data scientist that is not grounded, in some form or another, in some statistical principle or method. Concepts such as inference, modelling, and data visualization, are an integral part of the toolbox of the modern data scientist. Wild and Pfannkuch (1999) describe *applied statistics* as:

> "part of the information gathering and learning process which, in an ideal world, is undertaken to inform decisions and actions. With industry, medicine and many other sectors of society increasingly relying on data for decision making, statistics should be an integral part of the emerging information era".

A department that embraces *applied statistics* as defined above is a natural home for data science in academia. For a larger summary of the current discussions in the statistical literature describing how past contributions of the field have influenced today's data science, we refer the reader to Supplementary Section 1.

1.3 What is missing in the current Statistics curriculum? *Creating*, *Connecting, Computing*

Despite important subject matter insights and the discipline's applied roots, research in current academic statistics departments has mostly focused on developing general data analysis techniques and studying their theoretical properties. This theoretical focus spills over to the curriculum since a major goal is to produce the next generation of academic statisticians. Wild and Pfannkuch (1999) complained that:

> "Large parts of the investigative process, such as problem analysis and measurement, have been largely abandoned by statisticians and statistics educators to the realm of the particular, perhaps to be developed separately within other disciplines".

They add that "*[t]he arid, context-free landscape on which so many examples used in statistics teaching are built ensures that large numbers of students never even see, let alone engage in, statistical thinking*".

[2] We include biostatistics departments

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

While academic statisticians have made important theoretical and methodological contributions that are indispensable to the modern data scientist, the theoretical focus does not fully serve the needs of those searching for training in data science. In response to this, numerous authors have offered compelling arguments to expand and update the statistics curriculum emphasizing the importance of data science proficiency. For a discussion on expanding and updating the statistics curriculum, we refer the reader to Supplementary Section 2.

Here, we share our approach to developing and teaching a graduate-level, introductory data science course, which is motivated by an aspiration to emphasize the teaching of three key skills relevant to data science, which we denote as *creating*, *connecting*, and *computing*. These are skills that applied statisticians learn informally during their thesis work, in research collaborations with subject matter experts, or on the job in industry, but not necessarily in traditional courses. We explain each one below and later provide specific advice on approaches that can help students attain these skills.

Computing: The topic most discussed in opinion pieces about updating the curriculum is the lack of computing. For example, Nolan and Temple Lang (2012), Speed (2014), and Baumer (2015) all argue that there should be more computing in the statistics curriculum. These opinions ring true for us as researchers in the field of genomics in which computing is an essential skill. This is particularly the case for those of us that disseminate our statistical methodology by developing and sharing open source software (Huber et al. 2015).

Connecting: In our view, computing is not the main disconnect between the current statistics curriculum and data science. Instead, we contend the main problem is that traditional statistics courses have focused on describing techniques and their mathematical properties rather than solving real-world problems or answering questions with data. Data are often included only as examples to illustrate how to implement the statistical technique. While knowledge and understanding of the techniques are indispensable to using them effectively and to avoid reinventing the wheel, the more relevant skill pertaining to data science is to **connect** the subject matter question with the appropriate dataset and analysis tools, which is not currently prioritized in statistics curricula. This relates to what happens during what Wild and Pfannkuch (1999) refer to as the *interrogative cycle* of *statistical thinking*. In contrast, the typical approach in the classroom is to mathematically demonstrate that a specific method is an optimal solution to something, and then illustrate the method with an unrealistically clean dataset that fits the assumptions of the method in an equally unrealistic way. When students use this approach to solve problems in the real-world, they are unable to discern if the dataset is even appropriate for answering the question and unable to identify the most appropriate methodological approach when it is not spoon fed. Furthermore, knowing how to leverage application-specific domain knowledge and interactive data visualization to guide the choice of statistical techniques is a crucial data science skill that is currently relegated to learning on the job. For a much more detailed exposition on this idea, consult Wild and Pfannkuch (1999).

Creating: The current statistical curriculum tacitly teaches students to be passive: wait for subject-matter experts to come to you with questions. However, in practice data scientists are

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

often expected to be active; they are expected to **create** and to formulate questions. Note that many of the founders of the discipline of Statistics did just this. For example, as early as the 1830s, Quetelet created large demographic datasets to describe, for example, the association between crime and other social factors (Beirne 1987). Francis Galton collected father and son height data to answer questions about heritability (Galton 1886 and 1889). Ronald Fisher designed agricultural experiments and collected yield data to determine if fertilizers helped crops (Fisher 1921). Examples abound, but in today's academia this is much more common among data analysts in other departments such as Economics where, for example, data analysts are designing randomized controlled trials, which are increasingly being used to guide policy interventions[3].

1.4 Bridging the gap in the classroom to teach Data Science

While many have called for statisticians to rebuild the undergraduate curriculum (Cobb 2015) or that statistics departments build data science courses (Baumer 2015; Hardin et al. 2015), here, we add that the individuals tasked with bridging the gap must be experienced themselves in *creating*, *connecting* and *computing*. Adding data and computing to a standard course is not sufficient. It should not be assumed that once data appears, theoretical knowledge automatically translates into the necessary skill to build an effective course or curriculum. For the very same limitations we describe for the curriculum, a PhD in Statistics is not sufficient for becoming an effective Data Science educator. We therefore *encourage* applied statisticians experienced in creating, connecting and computing to become *involved* in the development of courses and updating curricula. Similarly, we encourage statistics departments to reach out to practicing data analysts, perhaps in other departments or from other disciplines, to collaborate in developing these courses.

To aid in bridging the gap, as applied statisticians that have taught data sciences courses, we synthesize our recommendations on how to prepare and teach a graduate-level, introductory course in data science. We share a set of guiding principles and offer a detailed guide on how to teach an introductory course to data science. This guide can also be used for statisticians wanting to gain more practical knowledge and experience in computing, connecting and creating before embarking on teaching a data science course. In the Discussion Section, we include thoughts and recommendations for more advanced or specific data science courses.

2. PRINCIPLES OF TEACHING DATA SCIENCE

The key elements of our course can be summarized into five guiding principles (Table 1).

2.1. Organize the course around a set of diverse case studies

The first and most fundamental change we made to the standard statistical course pertaining to data science was to bring the subject matter question to the forefront and treat the statistical techniques and computing as tools that help answer the question. Through answering these questions, students learn to connect subject matter to statistical frameworks and also learn new statistical techniques as part of the data analysis process. Only after the

[3] Poverty Action Lab (https://www.povertyactionlab.org/), Government Performance Lab (http://govlab.hks.harvard.edu/), and Education Innovation Laboratory (http://edlabs.harvard.edu/)

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

techniques are motivated and introduced as solutions to a problem, are the mathematical details and justifications described. We carefully and deliberately select a set of diverse case studies to give us an opportunity to motivate and teach the wide range of techniques needed to be a successful data scientist. We discuss specific case studies in Section 3.2.

This approach is certainly not new and was explicitly proposed in 1999 by Nolan and Speed in the book *Stat Labs*, which promotes teaching mathematical statistics through in-depth case studies (Nolan and Speed 1999 and 2000). The abstract of the book eloquently describes the main guiding principle we followed when developing the class:

> “Traditional statistics texts have many small numerical examples in each chapter to illustrate a topic in statistical theory. Here, we instead make a case study the centerpiece of each chapter. The case studies, which we call labs, raise interesting scientific questions, and figuring out how to answer a question is the starting point for developing statistical theory. The labs are substantial exercises; they have nontrivial solutions that leave room for different analyses of the data. In addition to providing the framework and motivation for studying topics in mathematical statistics, the labs help students develop statistical thinking. We feel that this approach integrates theoretical and applied statistics in a way not commonly encountered in an undergraduate text.”

In a related recent blog post[4], Jeff Leek describes the need to put the *problem first.* We note that in our experience selecting motivating case studies and developing an illustrative data analysis for each one required the largest time investment.

2.2. Integrate computing into every aspect of the course

Donald Knuth originally introduced the idea of literate programming, which weaves instructions, documentation and detailed comments in between machine-executable code producing a document that describes the program that is best for human understanding (Knuth 1984). Recently, educational and scientific communities have begun to adopt this concept in the classroom by increasing the use of computing and requiring student assignments be completed using literate programming (Baumer et al. 2014). Here, we advocate that the lectures in a data science course also be created with literate programming. This demonstrates to students what is expected in the homework assignments, stresses the importance of computational skills (Nolan and Temple Lang 2012), and maximizes the “hands-on” experience of data analysis in the classroom. Creating lectures with literate programming allows students to observe the code used by an expert, the instructor. Furthermore, it also permits the instructor to write code live and to explore data during lecture, which gives students an opportunity to see and to hear an expert’s rationale for the choice of statistical technique and computational tool.

The use of literate programming also promotes reproducible research (Claerbout 1994; Buckheit and Donoho 1995). In the classroom, this is useful not only for the student assignments and projects, which provides a practical learning experience, but also the

[4] http://simplystatistics.org/2013/05/29/what-statistics-should-do-about-big-data-problem-forward-not-solution-backward/

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

lectures, because it can alleviate a major frustration for students who might spend significant time outside of class trying to repeat the steps of an analysis presented in lecture that did not integrate the data, code and results in one document.

Finally, literate programming also facilitates the implementation of active learning approaches. Given that students need to absorb subject matter knowledge, statistical concepts, and computing skills simultaneously, and falling behind on any one can result in a student missing out key concepts, we find that regularly challenging students with assessments permits the lecturer to gauge the degree of understanding from the students and adapt accordingly. Section 3.3. provides more details on how we implemented this principle.

We note that computing is a rather complex topic within itself. Programming requires ways of thinking that are difficult to develop. In our course, we assume that students have some programming skills. These skills can be attained in a basic programming course including several free online courses.

2.3. Teach abstraction, but minimize reliance on mathematical notation

Abstraction in the context of data analysis is one of the statistical discipline's most important contributions. We agree with Wild and Pfannkuch that "*[t]he cornerstone of teaching in any area is the development of a theoretical structure with which to make sense of experience, to learn from it and transfer insights to others.*" Although we do not implement the standard approach to teaching abstraction, we do consider it a fundamental aspect of data science and therefore recommend teaching it. One way in which we differ from the standard approach is that we avoid mathematical notation when possible. Mathematical notation provides the most efficient way of describing statistical ideas. However, we find that it often obfuscates the teaching of fundamental concepts. So when possible, we use computational approaches, such as data visualization or Monte Carlo simulations (Cobb 2007; Hesterberg 2015; Horton 2015) to convey concepts. Minimizing mathematical notation also forces students to grasp the ideas in a more intuitive way. George Cobb refers to this as "seeking depth" or taking away what is technical (formalisms and formulas) to reveal what is fundamental (Cobb 2015). Furthermore, mathematical formalisms are easier to learn after concepts have been grasped.

2.4. Structure course activities to realistically mimic a data scientist's experience

Once a question has been formulated, the data scientist's experience can be described as an iterative and interactive process which involves learning about the subject matter, data wrangling, exploratory data analysis, and implementing or developing statistical methodology. Wild and Pfannkuch (1999) describe "s*tructured frameworks*" for this process using the *Problem, Plan, Data, Analysis, Conclusion* cycle which is "concerned with abstracting and solving a statistical problem grounded in a larger 'real' problem…. *Knowledge gained and needs identified within these cycles may initiate further investigative cycles*". In this process it is rarely clear what method is appropriate, nor is there an obvious, uniquely correct answer or approach. Mimicking this experience will greatly help students learn to **connect**. Homeworks should be designed to be open-ended and based on real-world case studies. Students should be given an opportunity to commence work before the

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

statistical techniques or software tools, considered to be appropriate by the instructor, are introduced. Lectures should demonstrate how to efficiently use **computing** in the data analysis process and homeworks give students time to practice and become proficient at that skill. In a final project, students can be asked to define their own question, which will help them learn to **create**.

2.5. Demonstrate the importance of critical thinking/skepticism through examples

As Tukey[5] once said "*[t]he combination of some data and an aching desire for an answer does not ensure that a reasonable answer can be extracted from a given body of data.*" Culturally, academic statisticians are cautious and skeptical. Today's data science culture is much more optimistic about what can be achieved through data analysis. Although we agree that overly pessimistic attitudes should be avoided as they can lead to missed opportunities (Speed 2014), we are mutually concerned that there is currently too much hubris. To that end, we would argue that it is indispensable that data science courses underscore the importance of critical thinking and skepticism. Being skeptical and cautious has actually led to many important contributions. An important example is how randomized controlled experiments changed how medical procedures are evaluated (Freedman, Pisani and Purves 1998). A more recent one is the concept of the false discovery rate (Benjamini and Hochberg 1995), which has helped reduce the reporting of false positives in, for example, high-throughput biology (Storey and Tibshirani 2003). Ignoring these contributions has led to analysts to being fooled by confounders, multiple testing, bias, and overfitting[6]. Wild and Pfannkuch (1999) point out that "[a]s you gain experience and see ways in which certain types of information can be unsoundly based and turn out to be false, you become more sceptical". We agree and the only way we know how to teach this is through illustration with real examples. For example, providing case-studies of Simpson's Paradox can illustrate how confounding can cause spurious associations (Horton 2015).

3. CASE STUDY: INTRODUCTION TO DATA SCIENCE

We developed and taught a graduate course titled Introduction to Data Science (BIO 260) at the Harvard T.H. Chan School of Public Health (HSPH) in Spring 2016 (http://datasciencelabs.github.io/). Here, we present this course as a demonstration with specific examples of how we implemented our principles of teaching an introductory course in data science.

3.1 Course logistics

We used the R programming language, but we have used Python in another similar course (http://cs109.github.io/2014/). We do not consider the choice between R and Python an important one, but we do find beginers have an easier time learning R, in particular when the R packages in the *tidyverse* (Wickham 2016), such as *dplyr* (Wickham and Francois 2016) and *ggplot2* (Wickham 2009), are used.

[5] http://www.azquotes.com/quote/603406
[6] http://simplystatistics.org/2014/05/07/why-big-data-is-in-trouble-they-forgot-about-applied-statistics/

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Due to the effort required to grade homeworks, which included open-ended questions, we had at least one TA per 15 students, but we would recommend an even higher TA to student ratio. This high TA to student ratio was key to the success of the course as grading open-ended homeworks was very time consuming. We recommend keeping the class small rather than pushing the limits on this ratio.

The course assumes that students have some programming experience and knowledge of statistics at an introductory level. There were no formal textbooks or reading requirements. Students had diverse educational backgrounds and interests (Figure 1). This diversity presented a challenge as most of the material was review for some and difficult for others. The high TA to student ratio helped those that needed review and having the lectures posted before class helped the students determine if they needed to attend the lecture that day. We also made videos of the lectures and code used in class available for students to review.

3.2 Case Studies

As described in Section 2.1, the most important feature of this course is that we brought the subject matter question to the forefront. We structured the course and demonstrated the entire data analysis process using a set of case studies (Table 2). We selected the case studies from diverse applications that would be of interest to our students. These case studies were also chosen to assure that a variety of techniques were motivated to be useful tools to answer the questions. These included summary statistics, exploratory data analysis, data wrangling, inference, modeling, regression, Bayesian statistics, databases, and machine learning. The "*Two Cultures"* as described by Breiman (2001) received equal time in the course between exploratory data analysis/visualization, inference/modeling, and machine learning. In Supplemental Section 3, we describe seven case studies that occupied most of the semester. Other topics we covered can be seen in detail on the course website (http://datasciencelabs.github.io/pages/lectures.html).

To prepare a given case study, we used the following steps: (1) Decide what concepts or skills (column 3 of Table 2) we want to cover. (2) Look for example questions that are of general interest (column 1 of Table 2). (3) Identify a public dataset (column 2 of Table 2) that we can use to answer the question. (4) Answer the question ourselves and summarize it into an RMarkdown report. (5) Determine if the solution is pedagogically acceptable (e.g. not too hard or too easy). (6) Separate material to be covered in class from material to be left for the homework. We note that not all examples selected in Step 2 satisfied all six steps and were thus discarded. For this reason, we had to consider several different examples in Step 2. We estimate that this process took an average of 40 hours per case study.

3.3 Integrate computing in all aspects of the course

Literate Programming: Using R Markdown and R pres—Each lecture and homework assignment was created using literate programming. We prepared lectures using R Markdown (Rmd) and R Presentations (Rpres) and rendered the presentations using RStudio, which provides functionality to easily convert from these formats to PDF or HTML (Xie 2015). More importantly, using RStudio also permitted us to run live data analysis during lecture. These documents were available on GitHub (https://github.com/

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

datasciencelabs/2016) to allow students to follow along and run code on their own laptops during class. For each lecture, there were three to four TAs available in the classroom who were walking around to answer questions in person. In addition, we included a link to a Google Document at the top of the R Markdown in each lecture to allow students a venue to ask questions if they did not want to interrupt the lecture. Note that in the course in which we used Python, we used Jupyter Notebooks which provide similar functionality to Rmd and Rpres. Karl Broman has provided several useful tutorials[7] in these formats and others.

Using active learning techniques to teach data science—We divided lectures into 10 to 30 minute modules and included 3–5 assessment problems in between. These questions consisted of multiple-choice or open-ended questions with most requiring a short data analysis. The solutions, in the form of code required to solve these assessments, were presented and discussed in class and added to the lectures only after the lecture was complete. We asked students to enter their answers in Google Forms that we created before lecture (Figure 2). Seeing these responses permitted us to adapt the pace of the lectures.

3.4 Structure course activities to realistically mimic a data scientist’s experience

Lectures and homeworks—Each lecture and homework was structured to mimic a data scientist’s experience by analyzing a data set as an in depth case study. Out of the six homework assignments, five were directly connected to the case studies discussed in the lecture. Students were provided the assignments prior to the corresponding lecture to familiarize themselves with the details of the real-world problem and promote initial thoughts and data explorations. During class, lectures began with a formal introduction to the real-world problem immediately followed by a discussion on how to **create** and actively formulate a question that can be answered with data as it relates to the problem at hand. Once a question had been formulated, the data, course material, code, and analysis tools were introduced and woven into the lecture in the order that they were needed to demonstrate the iterative and interactive data analysis process. This approach demonstrated to the students how to **connect** the subject matter question with the appropriate data and analysis tool. Furthermore, this approach emphasized the importance of **computing** because students could run the code provided in the lecture on their own laptops live in class.

Because the homework assignments were interconnected to the lectures (extensions of the same in depth case studies discussed in lecture), the students could immediately dive into the data analysis, answering both determinate and open-ended questions. The former provided a concrete and consistent way to grade the assignments and the latter promoted creative freedom in the choice of what techniques to apply and demonstrated the iterative data process, similar to a data scientist’s experience.

Midterm exams—During the majority of assessments, students worked with TAs and other students to complete assignments. To assure that each individual student was grasping basic concepts and gaining basic skills, we incorporated two midterm exams (one halfway and one at the end) in the course.

[7] http://kbroman.org/pages/tutorials.html

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

The Final Project—The students also completed a month long final project on a topic of their choice either on their own or in a group. This portion of the course most closely mimicked the data scientist's experience. The deliverables for the project included a project proposal, a written report of the data analysis in an R Markdown, a website, and a two-minute video communicating what the group learned. The project proposal described the motivation for the project, the project objectives, a description of the data, how to obtain the data, an overview of the computational methods proposed to analyze the data and a timeline for completing the project. TAs were paired together with 3–4 groups to meet to discuss the proposed projects and provide guidance. The students used the concepts learned in and outside of the course to complete the projects. Once the projects were complete, the submitted deliverables were reviewed and the best projects were highlighted at the end of the course.

3.5 Organizing Course Content with GitHub

We used git and GitHub repositories to store course content, build the course website, and organize homework submissions. GitHub provides a service, GitHub Pages (https://pages.github.com/), that facilitates the construction of the website (http://datasciencelabs.github.io/). The course content (https://github.com/datasciencelabs/2016) and data (https://github.com/datasciencelabs/data) were available in GitHub repositories (Figure 3). Note that on github we organized the material by concept and skill rather than by case study. We did this because it permitted students to look-up concepts during their final project.

Although we could have organized the course with a standard teaching tool such as Blackboard or Canvas, we chose to expose students to the notion of version control and achieved this by using one of the most popular systems, git, along with the web-based git repository hosting service, GitHub. GitHub is currently the most widely used resource for code developers including data scientists.

We spent one lecture introducing the concept of version control, git and GitHub. To demonstrate the mechanics, we created a test repository (https://github.com/datasciencelabs/test_repo) and asked all the students to use git to obtain a copy of this repository during the lecture. We also introduced the concept of making changes to local repositories and pushing the changes to remote repositories. After this lecture, students were able to stay in sync with the course repository to access the course material at the beginning of each lecture.

In addition, we used git and GitHub to create private repositories for students to submit their homework assignments (Figure 3). GitHub Classroom (https://classroom.github.com/) offers free private repositories to instructors, which can be used to create assignments and distribute starter code to students. Each student was given access to make changes to his or her own private repository to submit their homework assignment. The last commit was used as the final submission. The TAs were able to quickly and efficiently access and grade the homework submissions. Note that GitHub regularly offers new services so we recommend keeping up to date with the latest.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

4. DISCUSSION

We have presented a set of principles and a detailed guide for teaching an introductory data science course. The course was centered around a set of case studies and homeworks that permitted students an experience that closely mimicked that of a data scientist. The case studies were carefully chosen to motivate students to learn important statistical concepts and computing skills. Choosing these case studies, finding relevant data sources, and preparing didactic data analysis solutions required an enormous initial investment of time and effort. We recommend that academic institutions invest accordingly on faculty preparing these courses by, for example, increasing the percent effort associated with teaching these courses for the first time (Waller 2017). At institutions that can’t accommodate this, it may be necessary to form interdepartmental teams to assure that enough faculty teaching time is allocated to the course. Going forward, we will need new case studies and we propose that, as a community of educators, we join forces to create a collection of case studies.

We have taught a version of our introductory course twice: once as an upper-level undergraduate class at Harvard College and once as a graduate course in the School of Public Health. In both cases we considered the course to be a success. In both courses we had a diverse group of students in terms of their educational backgrounds, programming experience and statistical knowledge. We found that students were motivated and willing to do extra work to catch up in their areas of weakness. We attribute this partially to the fact that students were highly motivated to learn the material. Neither of our courses were requirements, thus all students chose to take the class. We do recognize that our population of students are likely above average in their adeptness to self learning. In fact, through the extension school, the course was offered to a more general population of students, and this population did find the course more challenging. In particular, they required more assistance with statistical concepts and learning Python/R. In future courses, we will set different requirements for different population of students. For example, graduate students will be expected to learn on their own, while extension school students will be required to know the programming language in advance and have taken an introductory statistics and probability course.

We do note that an even more introductory version of our class can serve as a first course in statistics. Exposing students to practical questions and data in the real-world can serve as great motivation for learning the concepts and mathematics presented in an introductory statistics course. We also note that our course is only an introduction and that several more advanced courses are necessary to meet the demand of those seeking data science training. Under the current makeup of undergraduate curricula, curricula for advanced students will need to flexible. For example, students with strong programming skills, but no statistical training, will benefit from courses teaching advanced probability, inference and methods, while statistics students with no computing experience may benefit from courses on optimization, algorithms and data structures.

We may also need to develop new courses, or adapt existing ones, that demonstrate real-world examples and applications of (1) Machine Learning, (2) software engineering courses that teach students the principles needed to plan, organize, code, and test data analysis

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

software, (3) advanced computing courses necessary to deal effectively with datasets that do not fit in memory and algorithms that require multiple compute nodes to be implemented in practice, and (4) subject-matter centered courses that expose students to the nuances of data science in the context of specific research areas such the social sciences, astronomy, ecology, education, and finance. Clearly, under current academic structures, data science programs will be best served by collaboration spanning more than one department. All students will benefit from courses putting a greater emphasis on data and data analysis. We are confident that our principles and guide can serve as a resource not just for specific new courses, but also to generally improve masters and PhD level curricula as well.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank Joe Blitzstein and Hanspeter Pfister, the creators of CS109, from which we borrowed several logistical ideas, David Robinson, our *tidyverse* and *ggplot2* guru, for advice and for his guest lectures, Alyssa Frazee who helped develop the movie ratings lecture, Joe Paulson for suggesting Google polls, Héctor Corrada-Bravo for advice on teaching Machine Learning, GitHub Education for providing free private repositories, Garrett Grolemund, Sherri Rose and Christine Choirat for presenting guest lectures, all of the TAs from our Introduction to Data Science course BIO260 (Luis Campos, Stephanie Chan, Brian Feeny, Ollie McDonald, Hilary Parker, Kela Roberts, Claudio Rosenberg, Ayshwarya Subramanian) and GroupLens for giving us permission to adapt and re-distribute part of the 'ml-latest' MovieLens data set on our course website. We thank Jeff Leek for comments and suggestions that improved the manuscript and Scott Zeger for a helpful discussion. Finally, we thank NIH R25GM114818 grant for partial support for creating the teaching materials.

References

Baumer B, Cetinkaya-Rundel M, Bray A, Loi L, and Horton NJ (2014), "R Markdown: Integrating A Reproducible Analysis Tool into Introductory Statistics," Technology Innovations in Statistics Education, 8.

Baumer B (2015), "A Data Science Course for Undergraduates: Thinking With Data," The American Statistician, 69, 334–42.

Beirne P. 1987 "Adolphe Quetelet and the origins of positivist criminology". American Journal of Sociology, 92, 1140–1169.

Benjamini Y, Hochberg Y (1995), "Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing," Journal of the Royal Statistical Society. Series B (Methodological), 57, 289–300.

Breiman L (2001), "Statistical Modeling: The Two Cultures," Statistical Science, 16, 199–231.

Buckheit JB, Donoho DL (1995), "WaveLab and Reproducible Research," In Wavelets and Statistics, edited by Antoniadis Anestis and Oppenheim Georges, 55–81. Lecture Notes in Statistics 103 Springer New York.

Claerbout J (1994), "Hypertext Documents about Reproducible Research," Tech. rep, Stanford University, Available at http://sepwww.stanford.edu/sep/jon/nrc.html.

Cobb GW (2015), "Mere Renovation Is Too Little Too Late: We Need to Rethink Our Undergraduate Curriculum from the Ground Up," The American Statistician, 69, 266–82.

Cobb GW (2007), "The Introductory Statistics Course: A Ptolemaic Curriculum?", *Technology Innovations in Statistics Education* 1, Available at https://escholarship.org/uc/item/6hb3k0nz.pdf.

Freedman D, Pisani R, Purves R (1998), Statistics, W. W. Norton & Company Inc New York, New York Chapter 1.

Fisher RA (1921), "Studies in crop variation. I. An examination of the yield of dressed grain from Broadbalk," Journal of Agricultural Science, 11, 107–135.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Galton F (1886), "Regression Towards Mediocrity in Hereditary Stature," Journal of the Anthropological Institute of Great Britain and Ireland, 15, 246–263.

Galton F (1889), Natural Inheritance, London, Macmillan.

Hardin J, Hoerl R, Horton NJ, et al. (2015), "Data Science in Statistics Curricula: Preparing Students to 'Think with Data'," The American Statistician, 69, 343–53.

Hesterberg TC (2015), "What Teachers Should Know About the Bootstrap: Resampling in the Undergraduate Statistics Curriculum," The American Statistician, 69, 371–86. [PubMed: 27019512]

Horton NJ (2015), "Challenges and Opportunities for Statistics and Statistical Education: Looking Back, Looking Forward," The American Statistician, 69, 138–45.

Horton NJ, Baumer B, Wickham H (2015), "Setting the Stage for Data Science: Integration of Data Management Skills in Introductory and Second Courses in Statistics," Chance, 28: 40–50. Available at http://chance.amstat.org/2015/04/setting-the-stage/.

Huber W, Carey VJ, Gentleman R, et al. (2015), "Orchestrating High-throughput Genomic Analysis with Bioconductor," Nature Methods, 12, 115–21. [PubMed: 25633503]

Knuth DE (1984), "Literate Programming," Computer Journal, 27, 97–111.

Nolan D, Temple Lang D (2012), "Computing in the Statistics Curricula," The American Statistician, 64, 97–107.

Nolan D, Speed TP (1999), "Teaching Statistics Theory through Applications," The American Statistician, 53, 370–75.

Nolan D, Speed TP (2000), Stat Labs: Mathematical Statistics Through Applications, Springer Texts in Statistics Springer New York.

Speed T (2014) "Trilobites and Us," Amstat News, American Statistical Association, 1 1, Available at http://magazine.amstat.org/blog/2014/01/01/trilobites-and-us/comment-page-1/.

Storey JD, Tibshirani R (2003), "Statistical significance for genomewide studies", Proceedings of the National Academy of Sciences, 100, 9440–9445.

Waller LA (2017), "Documenting and Evaluating Data Science Contributions in Academic Promotion in Departments of Statistics and Biostatistics". bioRxiv.10.1101/103093.

Wickham H (2009), Ggplot2: Elegant Graphics for Data Analysis. New York: Springer.

Wickham H (2016), "tidyverse: Easily Install and Load 'Tidyverse' Packages". R package version 1.0.0 Available at https://CRAN.R-project.org/package=tidyverse

Wickham H, Francois R (2016), dplyr: A Grammar of Data Manipulation. R package version 0.5.0.

Wild CJ, Pfannkuch M (1999), "Statistical Thinking in Empirical Enquiry," International Statistical Review, 67, 223–265.

Xie Y (2015), Dynamic Documents with R and Knitr, Second Edition Chapman & Hall/CRC The R Series. CRC Press.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

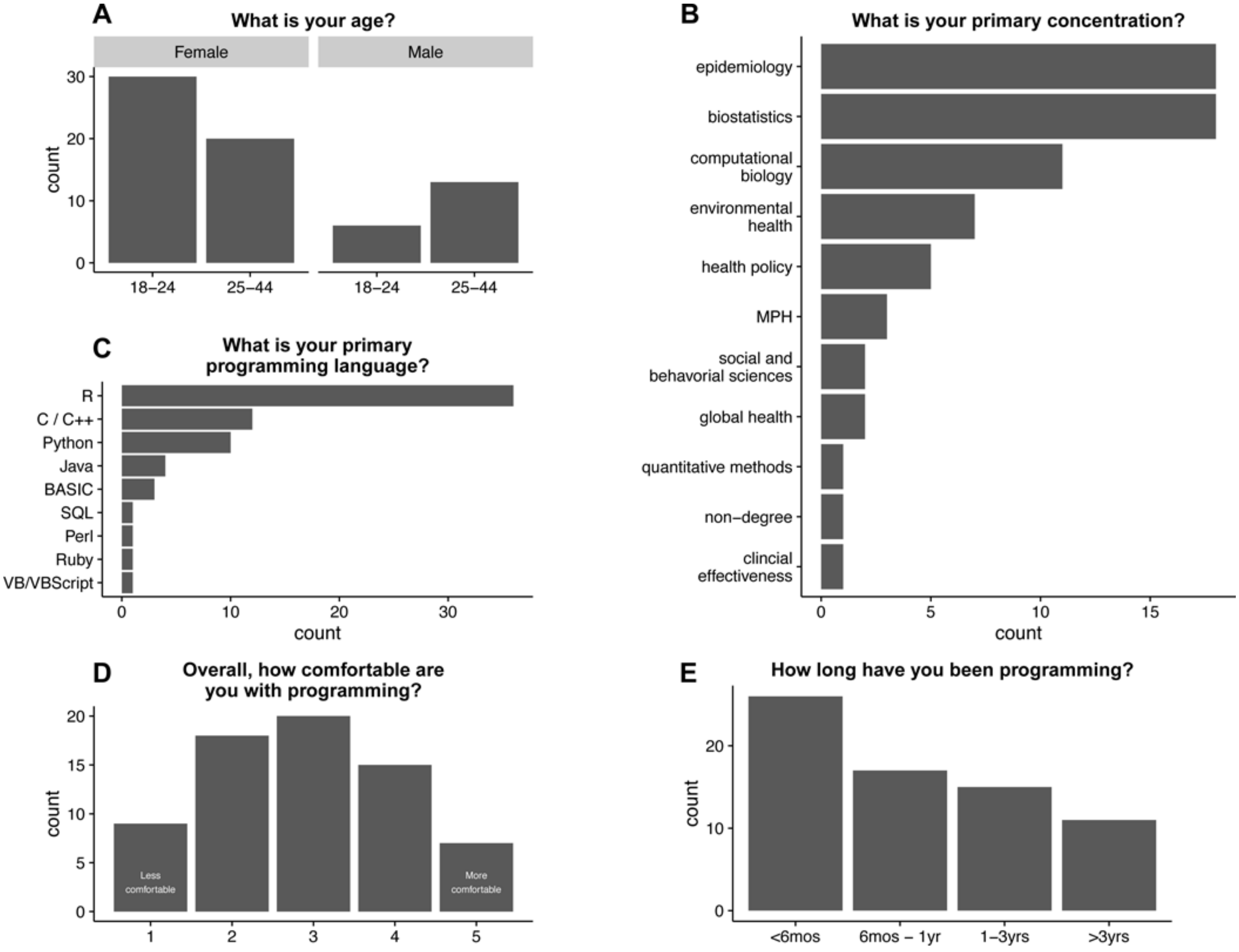


Figure 1.

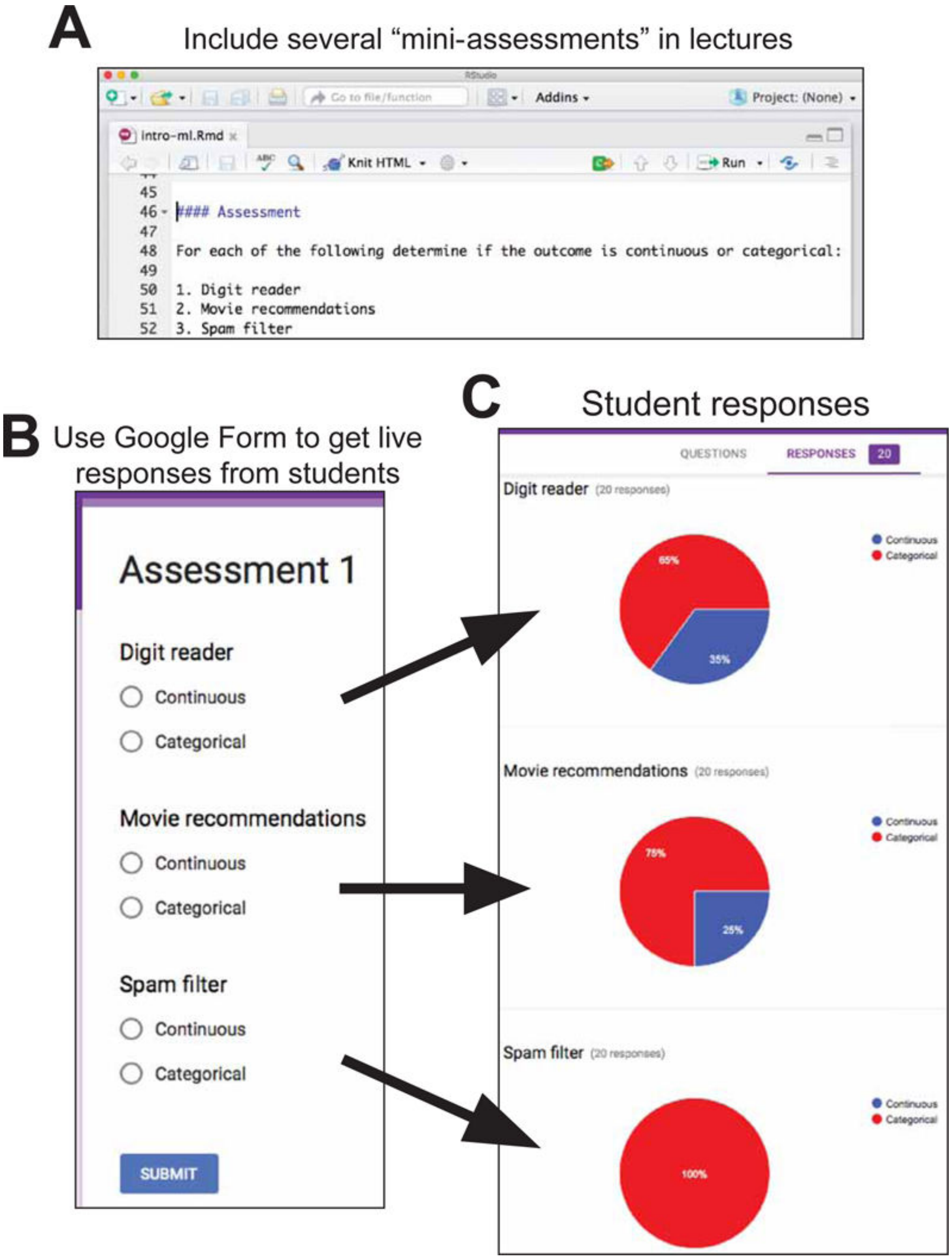
Summary from self-reported survey of students in the Introduction to Data Science (BIO 260) course taught at Harvard T.H. Chan School of Public Health in Spring 2016.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript



```
#### Assessment

For each of the following determine if the outcome is continuous or categorical:

1. Digit reader
2. Movie recommendations
3. Spam filter
```

Figure 2.

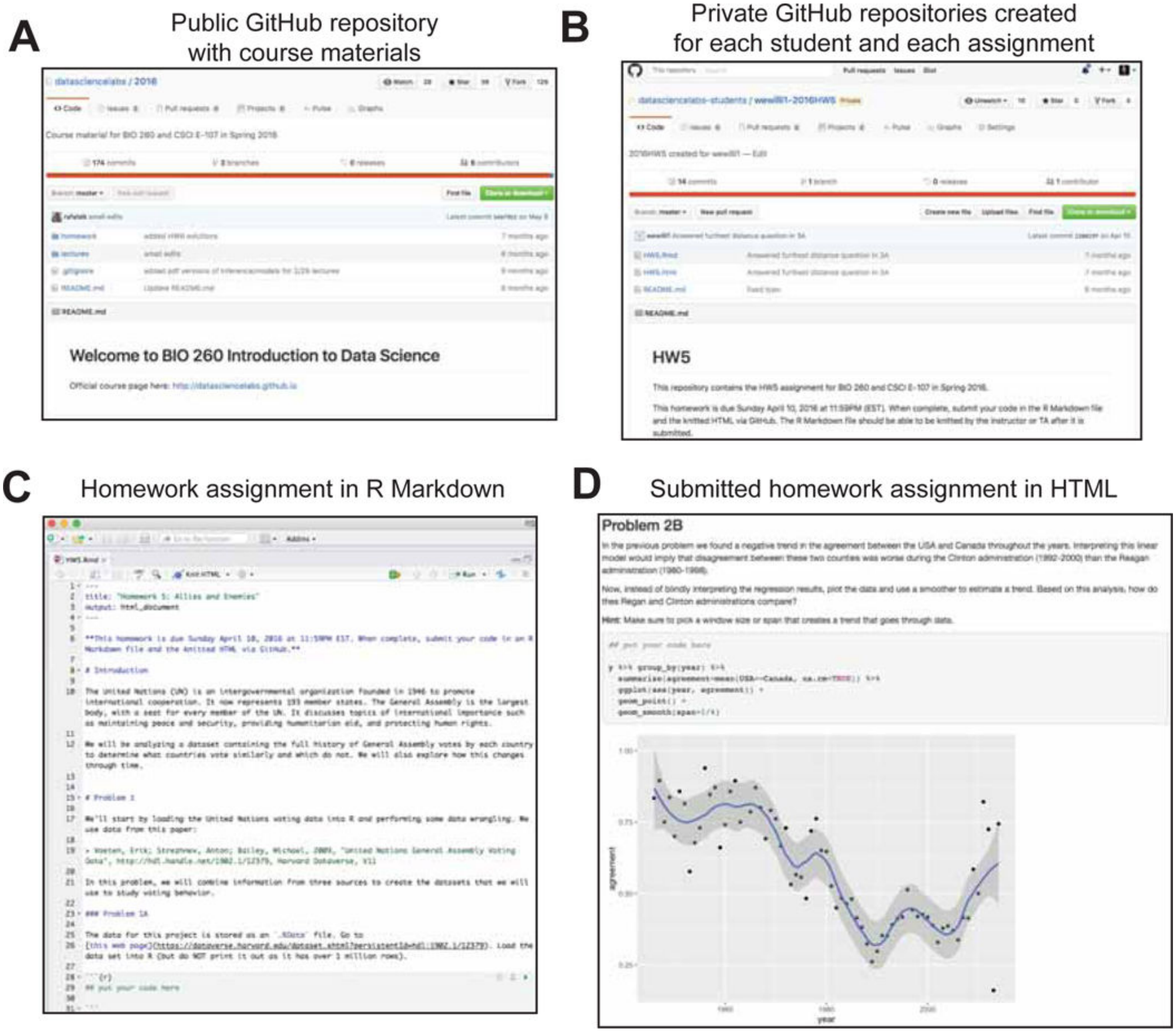
Using Google Forms as an active learning tool. (A) Three to five assessments were included in the R Markdown for each lecture, which consisted of either multiple-choice or open-ended questions. (B) Students were given a few minutes during the lecture to answer the questions. (C) Student responses were recorded and instructors could see the responses instantly. The live responses helped adapt the pace of the lectures.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript



Course material for BIO 260 and CSCI E-107 in Spring 2016

rafalab small edits — Latest commit on May 3

homework — added HW6 solutions — 7 months ago
lectures — small edits — 8 months ago
.gitignore — added pdf versions of inference/models for 3/29 lectures — 9 months ago
README.md — Update README.md — 8 months ago

README.md

Welcome to BIO 260 Introduction to Data Science

Official course page here: http://datasciencelabs.github.io

2016HW5 created for wewill81 — Edit

wewill81 Answered furthest distance question in 3A — Latest commit on Apr 10

HW5.Rmd — Answered furthest distance question in 3A — 7 months ago
HW5.html — Answered furthest distance question in 3A — 7 months ago
README.md — fixed typo — 8 months ago

README.md

HW5

This repository contains the HW5 assignment for BIO 260 and CSCI E-107 in Spring 2016.

This homework is due Sunday April 10, 2016 at 11:59PM (EST). When complete, submit your code in the R Markdown file and the knitted HTML via GitHub. The R Markdown file should be able to be knitted by the instructor or TA after it is submitted.

~~~
---
title: "Homework 5: Allies and Enemies"
output: html_document
---

**This homework is due Sunday April 10, 2016 at 11:59PM EST. When complete, submit your code in an R Markdown file and the knitted HTML via GitHub.**

# Introduction

The United Nations (UN) is an intergovernmental organization founded in 1946 to promote
international cooperation. It now represents 193 member states. The General Assembly is the largest
body, with a seat for every member of the UN. It discusses topics of international importance such
as maintaining peace and security, providing humanitarian aid, and protecting human rights.

We will be analyzing a dataset containing the full history of General Assembly votes by each country
to determine what countries vote similarly and which do not. We will also explore how this changes
through time.


# Problem 1

We'll start by loading the United Nations voting data into R and performing some data wrangling. We
use data from this paper:

> Voeten, Erik; Strezhnev, Anton; Bailey, Michael, 2009, "United Nations General Assembly Voting
Data", http://hdl.handle.net/1902.1/12379, Harvard Dataverse, V11

In this problem, we will combine information from three sources to create the datasets that we will
use to study voting behavior.

### Problem 1A

The data for this project is stored as an `.RData` file. Go to
[this web page](https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/12379). Load the
data set into R (but do NOT print it out as it has over 1 million rows).

```{r}
## put your code here

```
~~~

Problem 2B

In the previous problem we found a negative trend in the agreement between the USA and Canada throughout the years. Interpreting this linear model would imply that disagreement between these two counties was worse during the Clinton administration (1992-2000) than the Reagan admimistration (1980-1988).

Now, instead of blindly interpreting the regression results, plot the data and use a smoother to estimate a trend. Based on this analysis, how do thes Regan and Clinton administrations compare?

Hint: Make sure to pick a window size or span that creates a trend that goes through data.

```
## put your code here

y %>% group_by(year) %>%
  summarise(agreement=mean(USA==Canada, na.rm=TRUE)) %>%
  ggplot(aes(year, agreement)) +
  geom_point() +
  geom_smooth(span=1/4)
```

Figure 3.

Using git and GitHub to organize course materials and submit homework. (A) The course website was built with GitHub pages and a public course repository was created to incorporate the course material (lectures, homework, solutions and data). (B) Private GitHub repositories were created for each student and each homework assignment that contained starter code with the homework assignment. (C) Homework assignments were created in R Markdown and specific code chunks were created for the students to add their code as solutions. (D) Once the student was satisfied with their solutions, the homework submission was committed to the private GitHub repository as an R Markdown and HTML. The TAs were able to quickly and efficiently access and grade the homework submissions in the individual repositories.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 1.

A set of general principles of teaching data science meant to be used as a guide for individuals developing a data science course.

Principles of Teaching Data Science
• Organize the course around a set of diverse case studies
• Integrate computing into every aspect of the course
• Teach abstraction, but minimize reliance on mathematical notation
• Structure course activities to realistically mimic a data scientist’s experience
• Demonstrate the importance of critical thinking/skepticism through examples

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 2.

A table describing the case studies in our Introduction to Data Science course. Each case study is defined by a motivating problem or question, the data set used and the concepts and skills learned. We also provide the number of lectures (29 total) corresponding to the time spent on each topic.

Motivating question	Data set	Concepts/Skills learned	Number of lectures
How tall are we?	Self-reported heights of students in course	Introduction to R, R Markdown, RStudio, Exploratory data analysis (EDA), summary statistics	3
Are health outcomes and income inequality increasing or decreasing in the world?	Global health and economic data from Gapminder	Exploratory data analysis; Static and interactive data visualization using ggplot2	4
N/A	N/A		1
What are the basics of data wrangling?	Sleep times and weights for a set of mammals	Data wrangling using dplyr, tidyr	1
How do I get the course materials and submit homework?	N/A	Version control with git/GitHub	1

Who is going to win the 2016 Republican Primary?	U.S. Opinion polls from the 2008 and 2012 Presidential Election,
	2016 Presidential Primaries
		Probability, statistical inference Monte Carlo simulation;
		smoothing;
		Introduction to models, Bayesian statistics
			5

Midterm #1			1
How to pick players for a baseball team with a limited budget?	Sabermetrics baseball data	Regression; Data wrangling using broom	3
How to predict handwritten digits?	MNIST database of handwritten digits	Dimension reduction; Machine learning; Cross validation; Performance evaluation	4
How can I scrape data from a website?	Amazon book reviews for *ggplot2: Elegant Graphics for Data Analysis (Use R!)*	Scraping data from web Using rvest/CSS Selectors; regular expressions	1
How can I get data using an API?	Tweets with #rstats hashtag from Twitter	APIs; Sentiment analysis	1
Will I like this movie?	MovieLens user ratings of movies	Regularization; Matrix Factorization	1
Can we identify power plants within 10km of an air pollution monitor?	Power plants locations and measured air pollution	Structured Query Language (SQL); Relational databases	1
Is there gender bias in grant funding?	PNAS	Confounding; Simpson's Paradox	1
Midterm #2			1
Final Project Presentations			1